Modeling of Hurricane Impacts

Interim Report 1
March – May 2006

Dano Roelvink (P.I.), Ad Reniers,
Ap van Dongeren, Dirk-Jan Walstra

UNESCO-IHE Institute for Water Education (Contractor)
WL | Delft Hydraulics
Delft University of Technology

Contract no. N62558-06-C-2006

The Research reported in this document has been made possible through the support and sponsorship of the U.S. Government through its European Research Office of the U.S. Army.
Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
01-06-2006

2. REPORT DATE
Interim report

3. DATES COVERED *(From - To)*
March-May 2006

4. TITLE AND SUBTITLE
Modeling of
Hurricane Impacts,
Interim Report 1

5a. CONTRACT NUMBER
N62558-06-C-2006

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Roelvink, Dano
Reniers, Ad
van Dongeren, Ap
Walstra, Dirk-Jan

5d. PROJECT NUMBER

5e. TASK NUMBER
0001

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
UNESCO-IHE INSTITUTE FOR WATER EDUCATION
PO BOX 3015
2601 DA DELFT
NETHERLANDS

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
FISC SIGONELLA
NAVAL REGIONAL CONTRACTING DET LONDON
SHORE/FLEET TEAM
BLOCK 2, WING 12, GOVT BLDGS LIME GROVE
RUISLIP MIDDLESEX HA4 8BX
UNITED KINGDOM

10. SPONSOR/MONITOR'S ACRONYM(S)
ERDC

11. SPONSORING/MONITORING AGENCY REPORT NUMBER

12. DISTRIBUTION AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
Prepared in collaboration with WL | Delft Hydraulics and Delft University of Technology

14. ABSTRACT
This first interim report describes the development of dune erosion algorithms, based on an analysis of existing approaches; it describes the activities related to the ShoreCirc model required to make it suitable for modeling of the nearshore hydrodynamics during hurricanes, and it describes a newly developed model for inner surfzone, swash and overwash processes. Significant progress was made here and a set of Matlab routines providing much of the needed functionality is included for evaluation and testing. A novel approach for solving the time-varying wave action balance is applied, where, in contrast with existing surfbeat models which are averaged over directional and frequency spectrum, the wave directionality is maintained, which removes the need for a separate wave model to provide wave directions.

15. SUBJECT TERMS
Surf zone, swash, overwash, wave groups, wave propagation, morphology

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
unclassified	unclassified	unclassified	unclassified	15	Prof. Dano Roelvink
					19b. TELEPONE NUMBER *(Include area code)* +31 15 2151838

Contents

Abstract

Following the visit by Brad Johnson, Joe Gailani and Ty Wamsley in February 2006 and after further discussions the (at least short-term) line of approach has been to adopt ShoreCirc as the nearshore vehicle of development, and to investigate the possibilities to either extend SC to include overland flow, breaching etc, or to couple it with a model specifically for the really nearshore, swash and above.

Ap van Dongeren has since downloaded ShoreCirc and has compiled the code on Delft Hydraulics' computers. He has since tried to couple the code with the existing wave driver developed by Ozkan, but at this point the coupling still has bugs. He has been in contact with Brad Johnson at University of Delaware and Tuba Ozkan about these problems. He is planning a trip to Delaware in the coming months, to discuss problems and further development.

Dirk-Jan Walstra has been working with Jaap van Thiel de Vries on the problem of modelling the dynamics of dune foot retreat, which has issues such as whether the dune erosion just follows the transport of sand in the wet profile or whether it is necessary to include processes to explicitly get the sand out of the dune by some physical process, and how to model this. They have been developing a schematised Matlab environment to better test the concepts, without the burden of a big model system.

Dano Roelvink and Ad Reniers have been looking at the feasibility of developing a separate model for the really shallow, swash and dune area and have made tangible progress. They are developing a set of Matlab functions for short wave propagation, instationary shallow water equations, sediment transport and continuity equations that can be coupled in various ways and are designed to cope with extreme conditions such as encountered during hurricanes. Since length scales are short in terms of wave lengths and supercritical flow is happening a lot, the numerical implementation is mainly first order upwind, which in combination with a staggered grid makes it quite robust; to keep the code simple and make coupling and parallellization easier, we have used explicit schemes with automatic time-stepping, thereby guaranteeing stability.

The short wave propagation contains a newly-developed time-dependent version of HISWA, which solves the wave refraction and allows variation of wave action in x,y, time and over the directional space, and can be used to simulate the propagation and dissipation of wave groups; the nice thing compared to the existing surfbeat model is that a separate wave model to predict the mean wave direction is not needed, and that it allows different wave groups to travel in different directions. We have done some principle tests so far and are able to simulate surfbeats running up and over dunes. Ad Reniers has included full wave-current interaction in the short wave propagation.

A basic sediment transport module has been included which solves the 2DH advection-diffusion equation and produces total transport vectors, which can be used to update the bathymetry. The pickup function so far has been a very simple function; in the coming period a more realistic version, following Reniers et al (2004), will be implemented.

We have looked at the possibility of coupling such a model with Shorecirc, by first coupling it with itself, cutting the domain in half and exchanging information two ways. This goes without any problems, so we guess that coupling with Shorecirc will be doable as well.

The coming period will be mainly devoted to testing of the various components against analytical, laboratory and field test cases and to investigate the coupling with ShoreCirc.

1 Introduction

This report is the first progress report of the project ‘Modeling of Hurricane Impacts’, contract no. N62558-06-C-2006, which was granted by the US Army Corps of Engineers, Engineer Research and Development Center (ERDC), European Research Office and administered by FISC SIGONELLA, NAVAL REGIONAL CONTRACTING DET LONDON, SHORE/FLEET TEAM.

The project is being carried out by Prof. Dano Roelvink of UNESCO-IHE (Principal Investigator), Dr. Ad Reniers of Delft University of Technology and Dr. Ap van Dongeren and Dirk-Jan Walstra of WL | Delft Hydraulics.

Following the visit by Brad Johnson, Joe Gailani and Ty Wamsley in February 2006 and after further discussions the (at least short-term) line of approach has been to adopt ShoreCirc as the nearshore vehicle of development, and to investigate the possibilities to either extend SC to include overland flow, breaching etc, or to couple it with a model specifically for the really nearshore, swash and above. These activities, which fall under tasks 4 and 7 of the project proposal, were indicated as having top priority and have consequently received most attention in the past 3 months.

In the following we will outline in some detail the activities and results over this period. In Chapter 2 the activities related to ShoreCirc are explained. Chapter 3 gives an overview of the analysis of dune erosion mechanisms and their relation to nearshore profile development. Chapter 4 gives a description of the new model with working title XBeach. Chapter 5 outlines the proposed activities for the coming 3 months.

2 ShoreCirc adaptations

The Shorecirc model has been developed by the research group of the late Ib Svendsen of the University of Delaware. Dr. Ap van Dongeren was part of that research group and is one of the co-writers of the code. At Delft, he has implemented the time-dependent energy and roller equations (Van Dongeren et al., 2003) into the code. These extensions are not yet in the motherversion and will be implemented in this project in the next quarter. In fact, the extension of the energy equation as described above will be implemented.

Next the Shorecirc code (for the midfield of a few surfzone widths) will be coupled to the Xbeach model (very nearshore and dune) and to output of the SWAN model (far field model for quasi-linear wave propagation over areas of order 100 km).

3 Dune erosion mechanisms

3.1 Introduction

Within the Morphos project WL | Delft Hydraulics and TU Delft are working on the modeling of dune erosion. Although many empirical formulations/models exist we are focusing on process based approaches. The problem with dune erosion is that the modeling of the retreat of the upper beach and dune front requires additional formulations which cannot be based on the traditional transport modeling techniques. Various approaches have been developed over the years which are primarily based on an extrapolation of the offshore directed sediment transport in some "last wet" point (see e.g. Steetzel, 1993). The most advanced process based model that works with such an approach is the DurosTa model (Steetzel, 1993). This model has undergone extensive validation and has proven to yield reliable and robust predictions for a range of conditions. However, this model contains several novelties of which their importance to an accurate dune erosion retreat is unkown. Issues that require further investigation are the extrapolation method and the transport formulations. Here we will be discussing our findings on the extrapolation methods.

3.2 Extrapolation methods

As a first step we investigated the effect of various extrapolation methods. All extrapolation methods are based on a preset transport, S_{base}, which is constant in time. In total three methods were investigated. The first method is developed in this project and involves a direct scaling of the transport using the local water depth.

$$S_x(x) = S_{base}\left(\frac{h(x)}{h_{base}}\right) \tag{0.1}$$

The second method uses a vertical extrapolation which results in a dune erosion retreat where the shape of the dune is maintained.

$$S_x(x) = S_{base}\left(1 - \frac{z(x) - z_{base}}{z_{top} - z_{base}}\right) \tag{0.2}$$

in which z is the local profile, subscripts *base*: point for last wet point, *top*: indicates the position of the dune top.

The third method uses the DurosTa formulation (see Steetzel, 1993) for details.

Some prilimenary results for these extrapolation methods are shown in Figure 1 for a schematic profile in which the base transport was kept constant at 1 m^3/m^3. In this figure the profile development (top row), sedimentation-erosion (middle row) and the cross-shore sediment transport (bottom row) are shown in time. The bold lines indicate the initial distributions.

What becomes apparent from these results is that method 1 (left column) yields very similar predictions as the DurosTa method (right column). The steepening of the dune face is represented in both methods, whereas method 2 (middle column) results in a dune retreat with a constant shape.

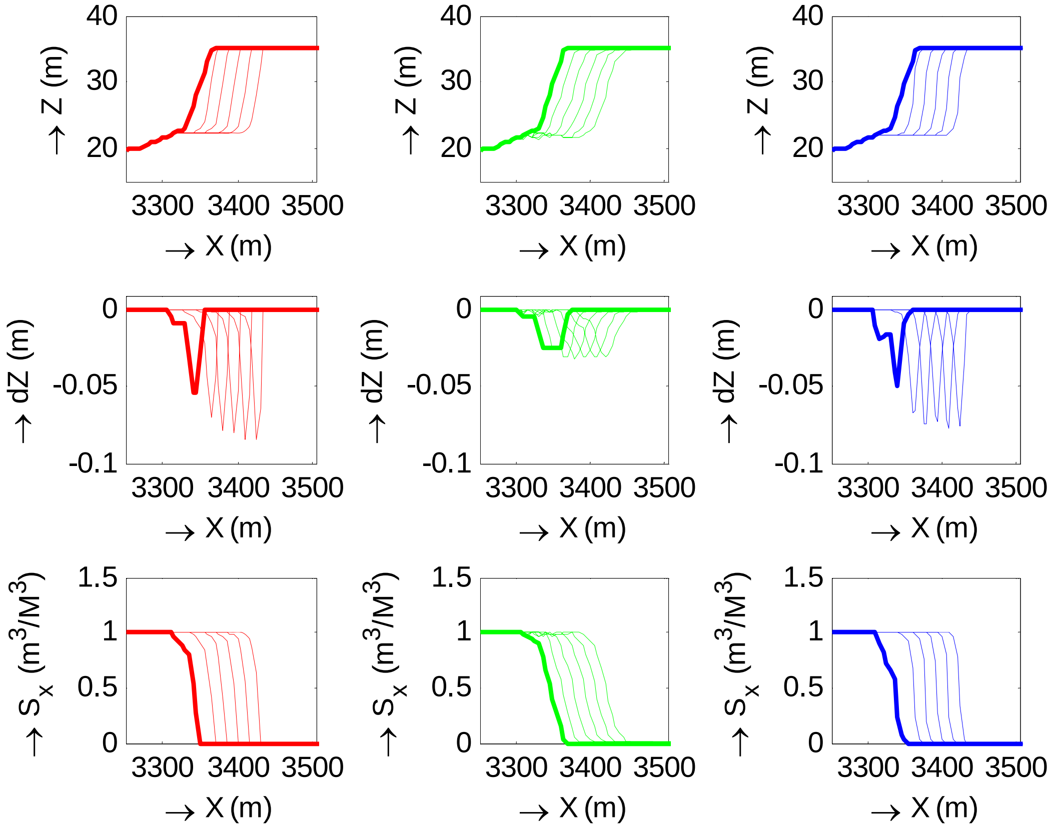

Figure 1 Dune erosion according to three different extrapolation methods. Top row: developmont of dune profile, second row: bottom changes per time step, third row: cross-shore sediment transport across dune. Left column: Method 1, middle column: Method 2, right column: Method 3.

3.3 Anticipated work

At the moment we are extending the work presented above by looking into the required numerical schemes and stability. Furthermore a detailed comparison between method 1 and DurosTa is required to assess the necessity of including many of the features that are available in the DurosTa model. In addition we will investigate various transport formulations and the effects of surfbeat on these types of extrapolation techniques.

4 XBeach model development

4.1 Objectives

The main objective of the XBeach model is to provide a robust and flexible environment in which to test morphological modelling concepts for the case of dune erosion, overwashing and breaching. The top priority is to provide numerical stability; first order accuracy is accepted since there is a need for small space steps and time steps anyway, to represent the strong gradients in space and time in the nearshore and swash zone. Because of the many shock-like features in both hydrodynamics and morphodynamics we choose upwind schematizations as a means to avoid numerical oscillations which can be deadly in shallow areas.

The modelling environment should be flexible and the code easy to comprehend and concise; therefore we have adapted the Matlab environment as development environment; conversion to Fortran 90/95 at a later stage will be straightforward.

The code should have the following functionality:

- Depth-averaged shallow water equations including time-varying wave forcing terms; combination of sub- and supercritical flows;
- Time-varying wave action balance including refraction, shoaling, current refraction and wave breaking;
- Wave amplitude effects on wave celerity;
- Depth-averaged advection-diffusion equation to solve suspended transport;
- Bed updating algorithm including possibility of avalanching;
- Possibility of online coupling with ShoreCirc;
- Possibility to extend to parallel multi-domain version;

4.2 General setup

The program XBeach consists of a main Matlab *script*, xbeach.m, and a number of *functions* that operate on two *structures*:

- par – this contains general input parameters
- s - this contains all the arrays for a given computational domain

For a single-domain run, one structure s is passed between flow, wave, sediment and bed update solvers, which extract the arrays they need from the structure elements to local variables, do their thing and pass the results back to the relevant structure elements. This makes the overall program clear, prevents long parameter lists and makes it easy to add input variables or arrays where needed.

For multi-domain runs, one can define multiple instantiations of the structure s which are passed to the same functions; an additional function is needed to pass the boundary information between the domains back and forth. We have carried out a simple test of this principle, without actually implementing a multi-processor version, which confirms that the data structure can handle this case.

In the Table 1 we will outline the various functions and their purposes. The main script xbeach.m is reproduced in Table 2 below.

Function call	Purpose
[par] = wave_input	Creates elements of structure par containing wave input parameters
[par] = flow_input(par)	Adds elements of structure par containing flow input parameters
[par] = sed_input(par)	Adds elements of structure par containing sediment input parameters
[s] = grid_bathy;	Creates grid and bathymetry and stores them in structure s
[s] = wave_dist (s,par);	Creates initial directional spectrum at sea boundary
[s,par]=wave_init (s,par);	Initialises arrays (elements of s) for wave computations
[s] = flow_init (s,par);	Initialises arrays (elements of s) for flow computations
[s] = sed_init (s);	Initialises arrays (elements of s) for sediment computations
[s] = wave_bc (s,par,it);	Wave boundary conditions update, each timestep
[s] = flow_bc (s,par,it);	Flow boundary conditions update, each timestep
[s]=wave_timestep (s,par);	Carries out one wave timestep
[s]=flow_timestep (s,par);	Carries out one flow timestep
[s]=transus(s,par);	Carries out one suspended transport timestep
[s]=bed_update(s,par);	Carries out one bed level update timestep
output(it,s,par)	Performs output

Table 1 Overview of Matlab functions XBeach

Of these functions, the input functions and grid_bathy are case-dependent at present and actually contain the input values and grid and bathymetry definitions. This can however easily be replaced by functions with the same output, which read data from input files or input them through a screen dialog. We aim to leave the initialisation, boundary conditions and computational functions case-independent. The output function can be adapted to fit specific needs.

```
clear all
%
% General input per module
[par] = wave_input;
[par] = flow_input(par);
[par] = sed_input(par);
% Grid and bathymetry
[s] = grid_bathy;
% Directional distribution wave energy
[s] = wave_dist (s,par);
% Initialisations
[s,par] = wave_init (s,par);
[s] = flow_init (s,par);
[s] = sed_init (s);

nt=par.nt;

for it=1:nt;
 %
    % Wave boundary conditions
    [s] = wave_bc (s,par,it);
    % Flow boundary conditions
    [s] = flow_bc (s,par,it);
    % Wave timestep
    [s] = wave_timestep (s,par);
    % Flow timestep
    [s] = flow_timestep (s,par);
    % Suspended transport
    [s]=transus(s,par);
    % Bed level update
    [s]=bed_update(s,par);
    % Output
    output(it,s,par)

end
```

Table 2 Main routine xbeach.m

4.3 Matlab code and test case

In the attached *xbeachv001.zip* the Matlab code is provided, for a test case of a plane sloping beach with a Gaussian hump superimposed, and longcrested wave groups are entering at a 20 degree angle to the shore normal. The grid, beach slope and hump dimensions can be easily modified in the function *grid_bathy.m* and incident wave conditions that drive this case are defined in *wave_input.m*.

In Figure 2 below a snapshot of the water level and velocity pattern is shown. The model can clearly handle wetting and drying and (breaking) long waves.

The details of the most important elements, the wave action equation solver and the shallow water equations solver, are given in the next sections.

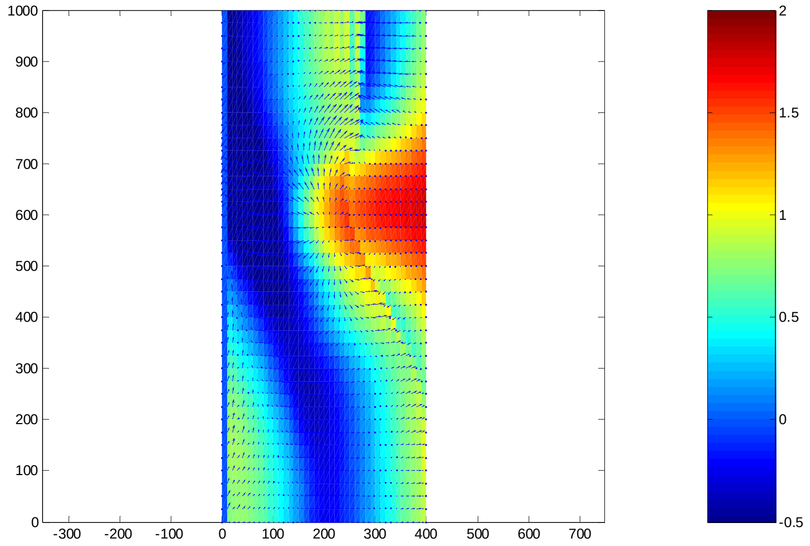

Figure 2 Snapshot of water level and velocity, beach-with-hump testcase.

4.4 Wave action equation solver

The wave forcing in the shallow water momentum equation is obtained from a time dependent version of the wave action balance equation. Similar to Delft University's HISWA model, the directional distribution of the action density is taken into account whereas the frequency spectrum is represented by a single mean frequency. The wave action balance is then given by:

$$\frac{\partial A}{\partial t}+\frac{\partial c_x A}{\partial x}+\frac{\partial c_y A}{\partial y}+\frac{\partial c_\theta A}{\partial \theta}=-\frac{D}{\sigma}$$

with the wave action:

$$A(x,y,\theta)=\frac{S_w(x,y,\theta)}{\sigma(x,y)}$$

where S_w represents the wave energy in each directional bin and σ the intrinsic wave frequency. The wave action propagation speeds in x- and y-direction are given by:

$$c_x(x,y,\theta)=c_g(x,y)\bullet\cos(\theta)+u(x,y)$$
$$c_y(x,y,\theta)=c_g(x,y)\bullet\sin(\theta)+v(x,y)$$

where θ represents the angle of incidence with respect to the x-axis. The propagation speed in θ-space is obtained from:

$$c_\theta(x,y,\theta)=\frac{\sigma}{\sinh 2kh}\left(\frac{\partial h}{\partial x}\sin\theta-\frac{\partial h}{\partial y}\cos\theta\right)+\cos\theta\left(\sin\theta\frac{\partial u}{\partial x}-\cos\theta\frac{\partial u}{\partial y}\right)+\sin\theta\left(\sin\theta\frac{\partial v}{\partial x}-\cos\theta\frac{\partial v}{\partial y}\right)$$

taking into account bottom refraction (first term on the RHS) and current refraction (last two terms on the RHS). The wave number k is obtained from the eikonal equations:

$$\frac{\partial k_x}{\partial t}+\frac{\partial \omega}{\partial x}=0$$
$$\frac{\partial k_y}{\partial t}+\frac{\partial \omega}{\partial y}=0$$

where the subscripts refer to the direction of the wave vector components and ω represents the absolute radial frequency. The wave number is the obtained from:

$$k=\sqrt{k_x^2+k_y^2}$$

The absolute radial frequency is given by:

$$\omega=\sigma+\vec{k}\bullet\vec{u}$$

and the intrinsic frequency is obtained from the linear dispersion relation:

$$\sigma=\sqrt{gk\tanh kh}$$

The group velocity is obtained from linear wave theory:

$$c_g=nc=\left(\frac{1}{2}+\frac{kh}{\sinh 2kh}\right)\frac{\sigma}{k}$$

This concludes the advection of wave action. The wave energy dissipation due to wave breaking is modelled according to Baldock et al. [1998]:

$$\bar{D}=\frac{1}{4}\alpha Q_b\rho g f_m\left(H_b^2+H_{rms}^2\right)$$

with $\alpha=O(1)$ and f_m representing the mean intrinsic frequency. The fraction of breaking waves is given by:

$$Q_b = \exp\left[-\left(\frac{H_b^2}{H_{rms}^2}\right)\right]$$

where the breaking wave height is:

$$H_b = \frac{0.88}{k}\tanh\left[\frac{\gamma kh}{0.88}\right]$$

and γ is a calibration parameter. The root mean square wave height is obtained from:

$$H_{rms} = \sqrt{\frac{8\int S_w(x,y,\theta)d\theta}{\rho g}} = \sqrt{\frac{8E_w}{\rho g}}$$

Next the total wave dissipation, $\bar{D}$, is distributed proportionally over the wave directions:

$$D(x,y,\theta) = \frac{S_w(x,y,\theta)}{E_w(x,y)}\bar{D}$$

This closes the set of equations for the wave action balance. Given the spatial distribution of the wave action and therefore wave energy the wave forcing can be calculated utilizing the radiation stress tensor:

$$F_x = -\left(\frac{\partial S_{xx}}{\partial x} + \frac{\partial S_{xy}}{\partial y}\right)$$
$$F_y = -\left(\frac{\partial S_{xy}}{\partial x} + \frac{\partial S_{yy}}{\partial y}\right)$$

And:

$$S_{xx} = \int\left(\frac{c_g}{c}\left(1+\cos^2\theta\right) - \frac{1}{2}\right)S_w d\theta$$
$$S_{xy} = S_{yx} = \int \sin\theta\cos\theta\left(\frac{c_g}{c}S_w\right)d\theta$$
$$S_{yy} = \int\left(\frac{c_g}{c}\left(1+\sin^2\theta\right) - \frac{1}{2}\right)S_w d\theta$$

Similar to the solution of the shallow water equations we use an up-wind schematisation to solve the wave action balance. The wave action is given at the same points at the water level. The advection of wave action is then discretized as follows:

$$\frac{\partial c_x^n A^n}{\partial x}(\mathrm{i,j,k}) = \frac{c_{x,i,j,k}^n A_{i,j,k}^n - c_{x,i-1,j,k}^n A_{i-1,j,k}^n}{x_{\mathrm{i,j}} - x_{\mathrm{i-1,j}}} \quad , c_{x,i,j,k}^n > 0$$

$$\frac{\partial c_x^n A^n}{\partial x}(\mathrm{i,j,k}) = \frac{c_{x,i+1,j,k}^n A_{i+1,j,k}^n - c_{x,i,j,k}^n A_{i,j,k}^n}{x_{\mathrm{i+1,j}} - x_{\mathrm{i,j}}} \quad , c_{x,i,j,k}^n < 0$$

$$\frac{\partial c_y^n A^n}{\partial y}(\mathrm{i,j,k}) = \frac{c_{y,i,j,k}^n A_{i,j,k}^n - c_{y,i,j-1,k}^n A_{i,j-1,k}^n}{y_{\mathrm{i,j}} - y_{\mathrm{i,j-1}}} \quad , c_{y,i,j,k}^n > 0$$

$$\frac{\partial c_y^n A^n}{\partial y}(\mathrm{i,j,k}) = \frac{c_{y,i,j+1,k}^n A_{i,j+1,k}^n - c_{y,i,j,k}^n A_{i,j,k}^n}{y_{\mathrm{i,j+1}} - y_{\mathrm{i,j}}} \quad , c_{y,i,j,k}^n < 0$$

$$\frac{\partial c_\theta^n A^n}{\partial \theta}(\mathrm{i,j,k}) = \frac{c_{\theta,i,j,k}^n A_{i,j,k}^n - c_{\theta,i,j,k-1}^n A_{i,j,k-1}^n}{\theta_{\mathrm{i,j,k}} - \theta_{\mathrm{i,j,k-1}}} \quad , c_{\theta,i,j,k}^n > 0$$

$$\frac{\partial c_\theta^n A^n}{\partial \theta}(\mathrm{i,j,k}) = \frac{c_{\theta,i,j,k+1}^n A_{i,j,k+1}^n - c_{\theta,i,j,k}^n A_{i,j,k}^n}{\theta_{\mathrm{i,j,k+1}} - \theta_{\mathrm{i,j,k}}} \quad , c_{\theta,i,j,k}^n < 0$$

Similar for the wave action balance:

$$\frac{A_{i,j,k}^{n+1} - A_{i,j,k}^n}{\Delta t} = -\left.\frac{\partial c_x^n A}{\partial x}^n\right|_{i,j,k} - \left.\frac{\partial c_y^n A}{\partial y}^n\right|_{i,j,k} - \left.\frac{\partial c_\theta^n A}{\partial \theta}^n\right|_{i,j,k} - \left.\frac{D}{\sigma}\right|_{i,j,k}$$

which yields the wave energy at the new time level.

4.5 Shallow water equations solver

Shallow water equations, neglecting Coriolis and horizontal diffusion terms, and (grey terms), for the moment, wind shear stress:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = \frac{\tau_{sx}}{\rho h} - \frac{\tau_{bx}}{\rho h} - g\frac{\partial \eta}{\partial x} + \frac{F_x}{\rho h} \tag{2.1}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = +\frac{\tau_{sy}}{\rho h} - \frac{\tau_{by}}{\rho h} - g\frac{\partial \eta}{\partial y} + \frac{F_y}{\rho h} \tag{2.2}$$

$$\frac{\partial \eta}{\partial t} + \frac{\partial hu}{\partial x} + \frac{\partial hv}{\partial y} = 0 \tag{2.3}$$

Here, *h* is the water depth, *u*, *v* are velocities in *x* and and *y* direction, τ_{bx}, τ_{by} are the bed shear stresses, g is the acceleration of gravity, η is the water level and F_x, F_y are the wave-induced stresses.

We apply an upwind schematisation, since the horizontal scale of the problem is limited and such a scheme deals with shocks in a natural way.

We apply a staggered grid, where bed levels and water levels are defined in the centre of cells, and velocity components at the cell interfaces.

If *nx,ny* are the number of cells in both directions, the water level points are numbered from 1 to *nx+1* and from 1 to *ny+1*.

The water level gradients are computed at the cell interfaces and are given by:

$$\frac{\partial \eta}{\partial x}(\mathrm{i,j}) = \frac{\eta_{\mathrm{i+1,j}} - \eta_{\mathrm{i,j}}}{x_{\mathrm{i+1,j}} - x_{\mathrm{i,j}}} \tag{2.4}$$

$$\frac{\partial \eta}{\partial y}(\mathrm{i,j}) = \frac{\eta_{\mathrm{i,j+1}} - \eta_{\mathrm{i,j}}}{x_{\mathrm{i,j+1}} - x_{\mathrm{i,j}}} \tag{2.5}$$

For computing the shear stresses at the cell interfaces we need the velocity magnitudes at these interfaces. These are composed by combining the normal velocity component at the interface and the average of the 4 adjacent tangential components:

$$\begin{aligned} v_{u,i,j} &= \frac{1}{4}(v_{i,j-1} + v_{i,j} + v_{i+1,j-1} + v_{i+1,j}) \\ u_{v,i,j} &= \frac{1}{4}(u_{i-1,j} + u_{i,j} + u_{i-1,j+1} + u_{i,j+1}) \end{aligned} \tag{2.6}$$

The water depth in each cell is computed as:

$$h_{i,j} = \eta_{i,j} - z_{b,i,j} \tag{2.7}$$

The depth at the interfaces is taken as the upwind depth in case the velocity is greater than a minimum velocity, or the average depth between the adjacent cells in case the velocity is less than this minimum velocity:

$$\begin{aligned} h_{u,i,j} &= h_{i,j} & &, u_{i,j} > u_{\min} \\ h_{u,i,j} &= h_{i+1,j} & &, u_{i,j} < -u_{\min} \\ h_{u,i,j} &= \frac{1}{2}(h_{i,j} + h_{i+1,j}) & &, \left| u_{i,j} \right| < u_{\min} \end{aligned} \tag{2.8}$$

$$\begin{aligned} h_{v,i,j} &= h_{i,j} & &, \ v_{i,j} > v_{\min} \\ h_{v,i,j} &= h_{i,j+1} & &, \ v_{i,j} < -v_{\min} \\ h_{v,i,j} &= \frac{1}{2}(h_{i,j} + h_{i,j+1}), & & \left| v_{i,j} \right| < v_{\min} \end{aligned} \tag{2.9}$$

The advection terms in x-direction are approximated as follows:

$$\begin{aligned} u\frac{\partial u^n}{\partial x}_{i,j} &= u^n_{i,j}\frac{u^n_{i,j} - u^n_{i-1,j}}{x^n_{i,j} - x^n_{i-1,j}} & &, u^n_{i,j} > 0 \\ u\frac{\partial u^n}{\partial x}_{i,j} &= u^n_{i,j}\frac{u^n_{i+1,j} - u^n_{i,j}}{x^n_{i+1,j} - x^n_{i,j}} & &, u^n_{i,j} < 0 \end{aligned} \tag{2.10}$$

$$v\frac{\partial u^n}{\partial y}_{i,j} = v^{\ n}_{u,i,j}\frac{u^n_{i,j+1} - u^n_{i,j-1}}{y^n_{i,j+1} - y^n_{i,j-1}} \tag{2.11}$$

The advection terms in y-direction are approximated as follows:

$$v\frac{\partial v^n}{\partial y}_{i,j} = v^n_{i,j}\frac{v^n_{i,j}-v^n_{i,j-1}}{y^n_{i,j}-y^n_{i,j-1}} \qquad ,v^n_{i,j}>0$$
$$v\frac{\partial v^n}{\partial y}_{i,j} = v^n_{i,j}\frac{v^n_{i,j+1}-v^n_{i,j}}{y^n_{i,j+1}-y^n_{i,j}} \qquad ,v^n_{i,j}<0 \tag{2.12}$$

$$u\frac{\partial v^n}{\partial x}_{i,j} = u^{\;n}_{v,i,j}\frac{v^n_{i+1,j}-v^n_{i+1,j}}{x^n_{i+1,j}-x^n_{i+1,j}} \tag{2.13}$$

The momentum equation is discretized as follows:

$$\frac{u^{n+1}_{i,j}-u^n_{i,j}}{\Delta t} = -u\frac{\partial u^n}{\partial x}_{i,j} - v\frac{\partial u^n}{\partial y}_{i,j} - \frac{g\,u^n_{i,j}\sqrt{u^{n\;2}_{i,j}+v^{n\;\;2}_{u,i,j}}}{h^n_{u,i,j}C^2} - g\frac{\eta^n_{\mathrm{i+1,j}}-\eta^{\mathrm{n}}_{\mathrm{i,j}}}{x_{\mathrm{i+1,j}}-x_{\mathrm{i,j}}} + \frac{F_{x,i,j}}{\rho h_{u,i,j}} \tag{2.14}$$

$$\frac{v^{n+1}_{i,j}-v^n_{i,j}}{\Delta t} = -v\frac{\partial v^n}{\partial y}_{i,j} - u\frac{\partial v^n}{\partial x}_{i,j} - \frac{g\,v^n_{i,j}\sqrt{u^{n\;\;2}_{v,i,j}+v^{n\;2}_{i,j}}}{h^n_{v,i,j}C^2} - g\frac{\eta_{\mathrm{i,j+1}}-\eta_{\mathrm{i,j}}}{y_{\mathrm{i,j+1}}-y_{\mathrm{i,j}}} + \frac{F_{y,i,j}}{\rho h_{v,i,j}} \tag{2.15}$$

From this, the velocities at the new time step level are computed. The water level is then updated by:

$$\frac{\eta^{n+1}_{i,j}-\eta^n_{i,j}}{\Delta t} = -\frac{u^{n+1}_{i,j}h^n_{i,j}-u^{n+1}_{i-1,j}h^n_{i-1,j}}{x_{\mathrm{u,i,j}}-x_{\mathrm{u,i\text{-}1,j}}} - \frac{v^{n+1}_{i,j}h_{\mathrm{i,j}}-v^{n+1}_{i,j-1}h^n_{i,j-1}}{y_{\mathrm{v,i,j}}-y_{\mathrm{v,i,j\text{-}1}}} \tag{2.16}$$

4.6 Conclusions and next steps

The Xbeach model is an as-simple-as-possible testing ground for algorithms related to wave propagation, shallow water flow including surfbeats, wetting and drying, sediment transport and morphological updating on shallow beaches and over dunes. With some Matlab experience it is easy to define grid, bathymetry and input conditions. Though the numerical scheme is rather simple and could undoubtedly be improved on specific points, it is quite robust and seems adequate for the moment.

In the coming months we will test the various parts of Xbeach against analytical, lab and field cases, which will be determined in consultation with the ERDC staff involved.